

AD-A052 891

SCIENCE APPLICATIONS INC MCLEAN VA

F/G 17/1

DOCUMENTATION OF PDP 11/70 SOFTWARE FOR DETERMINING TRAVEL-TIME--ETC(U)

N00039-76-C-0326

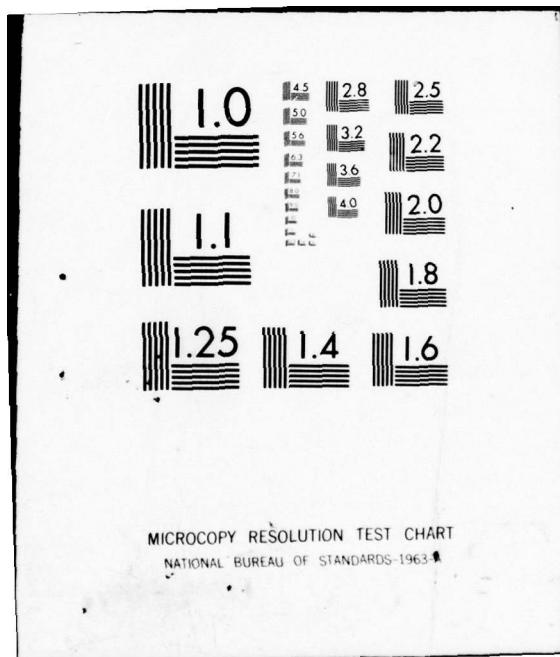
UNCLASSIFIED

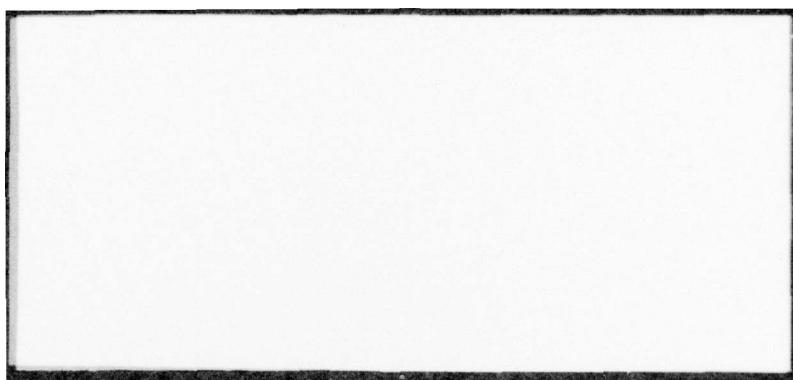
DEC 77 L S BLUMEN  
SAI-78-713-WA

NL

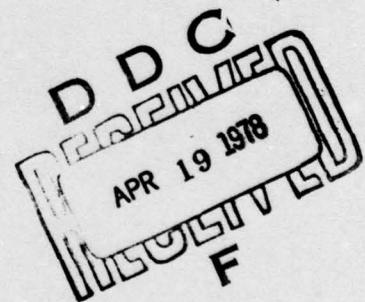
OF 1  
AD-A052891







Documentation of PDP 11/70 Software  
for Determining Travel-Time Differences  
from Cross Covariances



This document has been approved  
for public release and sale; its  
distribution is unlimited.



ATLANTA • ANN ARBOR • BOSTON • CHICAGO • CLEVELAND • DENVER • HUNTSVILLE • LA JOLLA  
LITTLE ROCK • LOS ANGELES • SAN FRANCISCO • SANTA BARBARA • TUSCON • WASHINGTON

⑥ Documentation of PDP 11/70 Software  
for Determining Travel-Time Differences  
from Cross Covariances

⑭ SAI-78-713-WA

⑪ 23 Dec. 1977

⑫ 44 p.

⑩ Prepared by L. S. Blumen

For NAVELEX 320

Under Contract No.: N00039-76-C-0326

SCIENCE APPLICATIONS INC.  
8400 Westpark Drive  
McLean, VA 22101  
Phone: 703 821-4470

408 404

ACCESSION for	
NTIS	White Section <input checked="" type="checkbox"/>
DDC	Buff Section <input type="checkbox"/>
UNANNOUNCED	
JSTOR	
Letter on file	
PY	
DISTRIBUTION/AVAILABILITY CODES	
S- CIAL	
X	

408 404

## Table of Contents

	<b>Page</b>
1.0 Introduction	1-1
2.0 Program Descriptions	2-1
2.1 Program NAV320	2-1
2.1.1 NAV2P	2-8
2.1.2 HANG	2-10
2.1.3 FIN	2-11
2.1.4 NAV34P	2-12
2.1.5 FILTR	2-13
2.1.6 DB	2-14
2.1.7 PWHITE	2-15
2.1.8 SHIFT	2-16
2.1.9 NAV78P	2-17
2.1.10 TRANS	2-18
2.2 Optional Plotting Capabilities	2-19
2.2.1 PLOT1	2-20
2.2.2 PLOT4	2-21
2.2.3 PLOT5	2-21
2.2.4 VPLOT	2-22
2.3 Optional Debug Capabilities	2-26
2.4 Operating Requirements	2-27

Section 1  
INTRODUCTION

→ The Navelex 320 Surface Shipping problem is resolved by the development of a method of correlating signals received from two widely spaced hydrophones, in an effort to locate the source. An exposition of the theoretical approach to this problem has been documented *in previously Ref. 1*. This report deals with the actual implementation of the solution, through the use of current computer technology. ←

A solution to the Navelex 320 Surface Shipping problem has been implemented in FORTRAN IV-plus on a DEC PDP 11/70 computer. Peripheral devices used consisted of a Floating Point Array Transform Processor (AP) and a Varian Statos line printer/plotter. The constraints imposed by the limited core size of the PDP 11/70, which has 32K of addressable core memory, have resulted in the development of a series of small, independent modules. The program driver and the main common blocks are the only elements that are continually core-resident. The other modules are stored on disk, and are rolled in from disk to core when necessary. These modules are returned to disk when their task has been executed. All intermediate storage of results is accomplished by reading from and writing to disk. In these ways, the size of the active modules in core at any one time has been minimized.

The following sections contain discussions of the individual modules and their functions, with the primary intention of facilitating conceptual understanding and usage of the program. The FORTRAN code is available upon request.

## Section 2

### PROGRAM DESCRIPTIONS

#### 2.1 Program NAV320 - Description of Flow

Program NAV320 is the driver for the NAVELEX surface shipping program. Its primary functions include the establishment of common arrays necessary to the execution of the program, the initialization of all starting values, and the relocation of the independent modules from disk storage to active core memory and back again when necessary.

The following variables are established for common usage in NAV320:

<u>Common block</u>	<u>Variable</u>	<u>Variable description</u>
AA	A(2050)	time series no. 1 consisting of 4 seconds of data in units of relative pressure
	B(2050)	time series no. 2; 4 seconds; in relative pressure
	NOC	processing index; indicates case no.
BB	N	number of samples (here 2048 for 4 sec)
	DBMED	tolerance level for smoothing option (dB)
	N2	N/2
NUMB	NUM	frequency shift index
NORMAL	SUMA	$A_i^2$ ; normalization factor
	SUMB	$B_i^2$ ; normalization factor

<u>Common block</u>	<u>Variable</u>	<u>Variable description</u>
START	NIN	internal flag indicating prior calculation and storage of Hanning function
	NIN2	number of records to skip before beginning processing; set internally
	NIN3	internal flag indicating prior calculation and storage of filter function

Initiation of a run requires user input. This input is supplied interactively, and through it the execution of optional capabilities is directed. The following information is requested of the user in an interactive mode:

#### PROCESSING INFORMATION

<u>Variable</u>	<u>Format</u>	<u>Variable description</u>
NA	22A1	name of data set which contains the A array e.g., SY:[xxx,yy]ppppppp.DAT where xxx,yy user identification no. pppppp file name
NB	22A1	name of data set which contains the B array; same format as NA
NSMO	I4	flag for smoothing option; 1 if smoothing desired
DBMED	F10.3	tolerance level (dB) for smoothing option

<u>Variable name</u>	<u>Format</u>	<u>Variable description</u>
NSTA	I4	number of starting record for array A
NSTB	I4	number of 1st record to use on data set that fills B
NFIL	I4	flag for filter option; =1 if filter desired
NT	I4	number of frequency shifts to perform on array A
DEL	F10.8	fractional increase in frequency bin width
NAVE	I4	number of times to average cross-spectral density
NO	I4	number of times to perform entire process

#### PLOTTING INFORMATION

<u>Variable name</u>	<u>Format</u>	<u>Variable description</u>
NCOMP	I4	=1 if only plot is to be a composite correlation plot; results will be written to a disk file for future plotting
NTSFLG	I4	=1 if plots of the original time series are desired
NPLTTS	I4	number of points to plot from original time series (.LE. 2048)
NSMFLG	I4	=1 if plots of smoothed amplitude are desired (only if smoothing option exercised)

<u>Variable</u>	<u>Format</u>	<u>Variable description</u>
NPLTSM	I4	number of points to be plotted of smoothed amplitude (.LE. 1023); required only if NSMFLG=1
NSHFLG	I4	=1 if phase plots of the shifted array (A) are desired; required only if (NT.ne.1 and NSMFLG.ne.1)
NHASE	I4	=1 if plots of phase corresponding to shifted amplitude desired; required only if (NSHFLG=1)
NPLTSH	I4	number of shifted amplitude or phase points to be plotted (.le. 1023)
FNAME	14A1	file name onto which composite correlation plot data will be written; same format as NA, NB
M	I4	=-1 if no single correlation plots desired =m for plots of every mth correlation

Program NAV320 sets up the major loop which directs execution of the surface shipping program. It is within this loop that the individual modules are invoked. The following is a description of the modules which are activated by NAV320:

<u>Module name</u>	<u>Formal Parameters</u>	<u>Description</u>	<u>Description of Module</u>
NAV2P	II	index indicating # of blocks averaged together (1-NAVE)	reads original time series A and B from data sets specified
	NSTA	start. loc. for A	
	NSTB	start. loc. for B	
	NUM	main driving index, (from 1-NO)	
PLOT1	NPLTTS	no. pts. to plot	plots original time series A and B
	INO	=1 for A =2 for B	
	IAMP	=0 for time series	
	N	array size (2048)	
NAV34P	none	-----	real FFT on A and B; normalizes by 1/2N
FILTR	none	-----	calculates and stores filter function for future use
DB	JDB	=0 I to dB =1 dB to I	conversion of arrays A and B from intensity to dB and back
PWHITE	ARRAY	array name	smoothes discrete discontinuities

<u>Module name</u>	<u>Formal Parameters</u>	<u>Description</u>	<u>Description of Module</u>
PLOT4	NPLTSM	no. pts. to plot	plots smoothed array A and B, amplitude and phase components
	INO	=1 for A =2 for B	
	IAMP	=1 for amp and phase plots	plots smoothed A and B, amplitude and phase components
	N	array size	
SHIFT	NT	number times to shift	forms frequency-shift
	DEL	fractional frequency shift	
PLOT5	NPLTSH	number points to plot	plots frequency shifted amplitude of A
	INO	(see above)	
	IAMP	=1 for amp and phase plots	
	N	array size	
NAV78P	NT	number of frequency shifts	multiples A*B <sup>*</sup> normalizes
	II	main index	
TRANS	NT	number of frequency shifts	FFT <sup>-1</sup> to yield correlation between two phones at each of NT frequency shifts
	NAVE	number of blocks of averages	

<u>Module name</u>	<u>Formal Parameters</u>	<u>Parameter Description</u>	<u>Description of Module</u>
PLOT9	NPLT	number points to plot	plots covariance of frequency shifted arrays
	INO	=1	
	IAMP	=0	
	N	number of points in array	

In addition, the following modules are invoked by other modules:

<u>Module name</u>	<u>Formal Parameters</u>	<u>Description</u>	<u>Description of Module</u>
HANG	none	-----	called by NAV2P to calculate and store Hanning function
FIN	IFLAG	error condition	called by NAV2P to terminate processing if error condition exists or normal end of processing has been reached
V PLOT	NPLT	number points to plot	driver for plotting routines; utilizes Varian software; called by the PLOTn routines
	INO	=1 for A =2 for B	
	IAMP	=0 for time series =1 for amplitude =2 for phase	
	N	array size	

The allocation of common blocks and their contents for each module may be seen in Tables 2.1 and 2.2. A flowchart, detailing the flow through the major components, is depicted in Fig. 2.1.

In the event that the user requests composite correlation plots, NAV320 causes creation of a file onto which all appropriate information is written immediately prior to termination of execution. This data file may then be accessed by the user at a later date for generation of composite correlation plots.

#### 2.1.1      Module NAV2P - Description of flow

Module NAV2P reads the original time series A and B from the data sets designated during the interactive session at the beginning of the run. The data represent the response at each of two hydrophones, in blocks of 2048 points, spanning a time period of 4 seconds. The data are in integer format, expressing relative pressure. NAV2P is called by the program driver (NAV320) in the following manner:

Call NAV2P(II,NSTA, NSTB, NUM)

where

II = index of averaged blocks

NSTA = starting record for A

**NSTB = starting record for B**

**NUM = driving index**

The two integer arrays are initially read by NAV2P. These are then converted to floating point arrays. The last two elements in the arrays, elements 2049 and 2050, are zero-filled (this is a requirement for the use of certain Varian plotter software routines and will be discussed in a subsequent section). These two arrays, both of size 2050, are then equivalenced to arrays A and B.

The status of the I/O process can be monitored through the value of a flag variable (IFLAG). The following are possible values of IFLAG:

**IFLAG =0 more data; no error**

**=1 end of file for data set filling array A**

**=2 end of file for data set filling array B**

**=3 end of file for both data sets**

**=4 error reading data set which fills array A**

**=5 error reading data set which fills array B**

If the value of IFLAG is other than 0 upon completion of execution of NAV2P, module FIN is called in the following manner:

```
CALL FIN(IFLAG,II,NUM)
```

Module FIN causes termination of execution and prints error messages when appropriate. FIN will be discussed in more detail in a subsequent section of this volume.

In the event that input data have been successfully read from the appropriate data files, module NAV2P issues a call to module HANG. HANG will calculate a Hanning function for a 2048 array. Module HANG will be discussed in further detail elsewhere in this volume. Once module HANG has been executed, NAV2P returns control to the main program driver.

#### 2.1.2 Module HANG - Description of flow

Module HANG establishes a Hanning function which will be applied to the 2048 points of arrays A and B. It is calculated in the following manner:

$$\text{Hanning weight}(i) = .5(1 - \cos(2\pi * (i-1)/N))$$

where N = no. of array points (here N=2048)

The Hanning function is actually calculated once, and is then stored for further usage.

HANG is called by NAV2P with no formal parameters and is executed entirely within the Array Transform Processor (AP). Initially, the Hanning function is calculated and stored within the AP. Next, arrays A and B are transferred from core into the AP. The A and B arrays are then multiplied by the Hanning coefficients in place. Normalization factors ( $\sum A_i^2$  and  $\sum B_i^2$ ) are calculated using a dot product AP library routine. The normalization factors and the Hanning weighted arrays A and B are returned from the AP to core memory. Module HANG then returns control to the main program driver.

#### 2.1.3 Module FIN - Description of flow

Module FIN is invoked by module NAV2P in the event that an error condition is identified by NAV2P during I/O operations that read arrays A and/or B from their original data sets. An error condition is identifiable by the value of the flag variable, IFLAG, which is passed to FIN by the call from NAV2P. If the value of IFLAG is any value other than 0 or 3, an error condition exists. This error may be the result of problems encountered while reading the individual data sets, or reaching asynchronous ends of file. Module FIN causes an appropriate error message to be printed on the console. The error message describes the nature of the error and gives its location in terms of the number of blocks and records that have been successfully read prior to encountering the error condition. In the event that the value of IFLAG is 3, processing has been successfully completed, and an appropriate message is printed on the console. FIN halts execution of the program in either case.

#### 2.1.4      Module NAV34P - Description of flow

NAV34P performs a real-to-complex Fast Fourier Transform (FFT) on the Hanning-weighted time series A and B, each containing 2048 real time-domain points. The results are normalized by a scale factor of  $(1/2N)$ . The resultant real and imaginary points are then converted into their corresponding amplitude and phase equivalents.

Module NAV34P is invoked by the program driver NAV320 with no formal parameters. All operations performed by NAV34P are executed within the AP. Initially, array A is moved from core memory to the AP. The real FFT is performed in place. Normalization, accomplished by vector-scalar multiplication, is then executed. A polar conversion replaces the real and imaginary values by their amplitude and phase counterparts.

At this point, the array is reordered to facilitate processing. The reordering process results in array A having the following structure:

D/C term	1023 amplitude terms	N/2 term	1023 phase terms
----------	----------------------	----------	------------------

1 D/C term (element 1)  
1023 amplitude terms (elements 2 - 1024)  
1 N/2 term (element 1025)  
1023 phase terms (elements 1026 - 2048)

Array A is then transferred from the AP back into core memory. Array B is brought into the AP, and the entire process is repeated for this second array. Once both arrays have been processed in this manner, NAV34P returns control to the main program driver.

#### 2.1.5      Module FILTR - Description of flow

Module FILTR calculates a filter function from a previously generated impulse response. The filter is applied to the frequency elements of arrays A and B. The first time routine FILTR is invoked, the filter function is calculated and stored in the AP for further use. On subsequent passes through this module, the frequency elements of arrays A and B are multiplied in place by the filter function components. The filter is a bandpass filter, which is applied to the amplitude components of arrays A and B, which are located in the first 1024 elements. As presently designed, the filter has a passband of 5 to 115 Hz.

A call to subroutine FILTR by NAV320 (no formal parameters necessary) causes arrays A and B to be transferred into the array transform processor from core. A check is then made to determine whether or not the filter function has been previously calculated and stored. If not, the impulse function is moved into the array transform processor, a real to complex FFT is performed on the impulse function, the results normalized and then moved into position for a vector multiply. Once the calculation of the filter function has been

performed, arrays A and B are multiplied, in place, by the filter function. Arrays A and B are then returned to core memory, and module FILTR returns control to the main program driver.

#### 2.1.6 Module DB - Description of flow

Module DB is called by NAV320 with one formal parameters to convert the amplitude elements in arrays A and B from and to dB units. DB is used in the event that the smoothing option is exercised. Before encountering module PWHITE, arrays A and B contain elements expressed in amplitude units. Subroutine DB converts these to dB units, using the following conversion rule:

$$A(\text{dB}) = 20 \log_{10}(A(\text{amplitude}))$$

Once this conversion has been performed, module PWHITE can proceed with its smoothing function, which is performed in dB units. At the conclusion of execution of PWHITE, module DB is invoked again, this time to convert the elements in arrays A and B back to amplitude units, in the following manner:

$$A(\text{amplitude}) = 10^{(A(\text{dB})/20)}$$

When subroutine DB has performed its conversion, control is then returned to the main program driver.

### 2.1.7      Module PWHITE - Description of flow

Ship self-noise in general consists of both broad-band and discrete line components. Module PWHITE filters out the discrete amplitude components of arrays A and B. PWHITE is invoked by the program driver NAV320 in the following manner:

CALL PWHITE (array)

where

array is the name of the array to be analyzed

Prior to calling PWHITE, NAV320 issues a call to DB. Module DB converts the elements of arrays A and B from amplitude to dB units (for a detailed discussion of module DB, refer to Section 2.1.6).

PWHITE utilizes a continuous sliding window, twenty-one elements in width, to filter the spectra. Discrete components are identified by comparison of the center element in the window with the median value for the window. If the value of the element is within a specified tolerance level of the median, the point remains unchanged. If not, its value is set equal to the median value for that window of twenty-one points. The window is then shifted one location, and the process is repeated until the entire array has been analyzed and all discrete components located and removed in this manner. Once PWHITE has analyzed arrays A and B, control is returned to the program driver, NAV320. NAV320 then invokes module DB, which converts the elements in arrays A and B back to amplitude units.

### 2.1.8      Module SHIFT - Description of flow

The signals received at widely spaced hydrophones from a moving overhead source will exhibit different Doppler shifts. For the case involving two hydrophones, it is necessary to adjust one of the spectra in order to eliminate the relative Doppler shift. This adjustment to the frequency scale is accomplished by linear interpolation of the amplitude and phase components of the spectrum. The theoretical development of this process is dealt with in greater detail in Ref. 1.

Module SHIFT performs a frequency shift on the elements of array A. SHIFT is invoked by NAV320 in the following manner:

CALL SHIFT(NT,DEL)

where

NT = # of frequency shifts to be performed

DEL =  $\delta_{\min}$ . the user-specified minimum fractional frequency shift

Initially, the A array and a variable which indicates the number of shifts to be performed on the array are brought into core memory from disk, where they have been stored temporarily. In the event that no frequency shifting is to be performed, the unshifted array is written to a summary disk file. For the case where frequency-shifting is to be performed, the size of the frequency shift is determined for each element in the array. The frequency shift for the  $i^{\text{th}}$  array element may be expressed as:

$$\delta_i = \delta_{\min}(i-1)$$

where

$\delta_{\min}$  is the user-specified minimum fractional frequency shift

A linear interpolation in amplitude and phase for adjacent points is then performed to yield frequency shifted values, which are then written to the summary disk file. This process is repeated until the requisite number of frequency shifts for the given array have been executed. At this point, module SHIFT returns control to the program driver, NAV320.

#### 2.1.9      Module NAV78P - Description of flow

Module NAV78P takes each of the NT frequency-shifted A arrays, multiplies the amplitude components of the A array by the amplitude components of the associated B array, finds the difference of the phase components of the two arrays, and converts the resultant array to its real and imaginary equivalent. The resultant array is then normalized by a factor of  $N/(SUMA*SUMB)^{1/2}$  and these NT arrays are written to disk for temporary storage. On subsequent passes (from 1 to NAVE) through NAV78P, the NT normalized arrays are added to the NT arrays already written to disk. This results in NT arrays, each of the arrays being the sum of the NAVE blocks which have been added together.

NAV78P operates primarily within the array transform processor. Initially, the B array is brought into the AP. A normalization factor is then calculated and stored in the AP. Then, for each of the NT frequency shifted A arrays, the process described in paragraph 1 is executed. If the current pass through NAV78P is the first time through, the results, which have been stored in array A, are written out to disk. If this is not the first pass through NAV78P, the calculated array is added to the previously calculated sums and then written to disk for temporary storage. This process is repeated until NAVE blocks have been processed in this manner. Upon completion of its task, module NAV78P returns control to the main program driver.

#### 2.1.10 Module TRANS - Description of flow

Module TRANS reads each of the NT array sums from disk, places them in the AP, and divides each value by NAVE, the number of arrays summed in module NAV78P. A complex to real FFT is performed on each array, yielding the cross covariance of signals received at the two hydrophones for each of the NT frequency shifts. As each covariance function is generated, it is written to disk for temporary storage. Once all NT covariance functions have been so processed, TRANS returns control to the main program driver. The call to TRANS is of the form:

**CALL TRANS(NT,NAVE)**

**where**

**NT**      **number of frequency shifts**

**NAVE**    **number of 2048 blocks to be averaged together**

## **2.2      Optional Plotting Capabilities**

At various stages throughout the execution of the program, it is possible for the user to request plots of particular functions. The NAVELEX program is currently equipped with the capability to produce any or all of the following plots:

- a) Original time series**
- b) Smoothed amplitude components and associated phase components**
- c) Frequency-shifted amplitude components and corresponding phase components**
- d) Final covariance functions**

These plots may be requested by the user during the interactive input phase at the beginning of each run. At that time, other information, pertinent to the composition of the requested plots, is required of the user as input. For a full explanation of all mandatory and optional input parameters, see Section 2.1, which discusses the main program driver, NAV320.

The flow of information for the plotting components is schematically represented in Fig. 2.2. For each of the aforementioned plotting capabilities, a module designated PLOTn, where n=1,4,5, or 9, is invoked by NAV320. These modules generate information (such as titles, axis labels, scale and displacement factors), used by the Varian subroutines which generate the plots. Each of the PLOTn routines, in turn, issues a call to module VPLOT, which positions pointers to the information to be plotted, and invokes the Varian subroutines. The Varian subroutines perform the actual plotting. The information generated by the PLOTn routines is passed to the other components through the use of common blocks as shown in Table 2.2.

#### 2.2.1      Module PLOT1 - Description of flow

Module PLOT1 is called by Program NAV320, the main program driver. PLOT1 is invoked when plots of the original time series A and B are desired. The function of PLOT1 is to set up the appropriate titles, axis labels, scale and displacement factors and other indices which allow the time series plots to be generated and labeled. Once these functions have been executed, PLOT1 issues a call to VPLOT, the module which invokes the Varian Statos line printer/plotter. Module VPLOT will be discussed in more detail in a subsequent section of this volume. When execution of VPLOT is completed, subroutine PLOT1 returns control to the main program driver. The call to PLOT1 is made in the following manner:

**CALL PLOT1(NPLT,INO,IAMP,N)**

<b>where</b>	<b>NPLT</b>	<b>no. of points to be plotted</b>
	<b>INO</b>	<b>=1 if array A is to be plotted</b> <b>=2 if array B is to be plotted</b>
	<b>IAMP</b>	<b>=0 if time series is to be plotted</b> <b>=1 if amplitude is to be plotted</b> <b>=2 if phase is to be plotted</b>
	<b>N</b>	<b>no. of points in original time series</b>

### **2.2.2 PLOT4 - Description of flow**

Module PLOT4 sets up titles, axis labels, scale and displacement factors, and other information necessary to the plotting of amplitude which has been smoothed by module PWHITE. All titles and axis labels are placed in arrays which consist of real words, 4 bytes long. Once PLOT4 has performed its functions, it issues a call to module VPLOT, which performs the actual plotting. This done, control is returned to the driver program.

### **2.2.3 Module PLOT5 - Description of flow**

The primary function of module PLOT5 is to initialize the titles, axis labels, scale and displacement factors, and other information necessary to the plotting of the frequency-shifted amplitude and phase components of array A, or the plotting of the unshifted amplitude

and phase components of array B. Initially, the frequency-shifted array to be plotted is read in from disk, where it has been temporarily stored. The components of the array are converted to dB units for plotting purposes, and the resultant arrays are normalized. Appropriate titles, etc. are then initialized, and PLOT5 then issues a call to module VPLOT, which initiates the actual plotting of the arrays. Once this has been accomplished, module PLOT5 returns control to the main program driver.

#### 2.2.4 PLOT9 - Description of flow

Module PLOT9 sets up titles, axis labels, the scale and displacement factors, and other information necessary to the plotting of covariances. The array to be plotted is read from disk, where it has been temporarily stored. Once this has been done, PLOT9 issues a call to module VPLOT, which performs the actual plotting of the covariance functions. When the covariance has been plotted, PLOT9 returns control to the main program driver.

#### 2.2.5 Module VPLOT - Description of flow

Module VPLOT is responsible for the actual plotting of original time series, amplitude, phase, and/or covariances. The PLOTn routines (where n=1,4,5, or 9) initialize all information necessary for a particular plot. Once this has been done, the PLOTn routines issue a call to VPLOT in the following manner:

CALL VPLOT(NPLT,INO,IAMP,N)

where

NPLT	no. of points to be plotted
INO	=1 to plot array A
	=2 to plot array B
IAMP	=0 to plot original time series
	=1 to plot amplitude and phase
	=2 to plot phase
N	no. of points in the original time series

Initially, VPLOT determines the value of the variable IAMP. This determines the type of data which are to be plotted (i.e., times series, amplitude, phase, or covariance). This, in turn, determines the starting locations and the maximum number of data points that may be plotted.

Plotting is accomplished through the use of the Varian Statos software, which is resident in the Dataplot library. All routines resident in the Dataplot library are invoked through FORTRAN subroutine calls to selected routines which perform a variety of utility functions.

The following is a list of the Varian subroutines called by VPLOT:

<u>Subroutine</u>	<u>Parameters</u>	<u>Param. desc.</u>	<u>Sub. desc.</u>
SOPEN	IWORK	start loc of work area	initializes all machine-dependent parameters;
	ISZ	size of work area	opens all LUNs
	IDSC	disk sector size	

<u>Subroutine</u>	<u>Parameters</u>	<u>Param. desc.</u>	<u>Sub. desc.</u>
ORIG	X	inches to off-set new origin from old	establishes new origin for current plot
	Y	inches to off-set new origin from old in Y direction	
SCALE	ARR	name of array to be scaled	scales data to fit on a page of length PGSZ; calculates scale factor and displacement factor which are placed in positions ARR(NPTS+1) and ARR(NPTS+2), respectively
	NPTS	no. of points in array to be scaled	
	PGSZ	length of page (in.)	
	INC	increment at which array is to be sampled	
CURR	X1	x-coord of 1st point	initializes first point of a continuous series of points
	Y1	y-coord of 1st point	
VECT	X1	x-coord of last point	creates a vector from (X1,Y1) to (X2,Y2)
	Y1	y-coord of last point	
	X2	x-coord of current point	
	Y2	y-coord of current point	
CHAR	X	x-coord of 1st letter	prints title for plot
	Y	y-coord of 1st letter	
	HT	height of each letter	
	IBCD	address of 1st word to be plotted	

<u>Subroutine</u>	<u>Parameters</u>	<u>Param. desc.</u>	<u>Sub. desc.</u>
CHAR cont'd.	ANGC	angle char. makes with X-axis	
	ANGS	angle char. string is to make with X-axis	
	NCHAR	no. of charac- ters to be plotted	
AXIS	X	x-coord of ori- gin relative to default origin	draws an axis and labels values in 1-inch inter- vals
	Y	y-coord of ori- gin relative to default	
AXLH	AXLH	length of axis	
	ANG	angle axis is to make with direction of paper movement	
IBCD		address of char. string to use as label for axis	
	NCHAR	no. chars. in char. string	
	VLO	value at origin	
SF	SF	increment between tick marks	
PLOT	ICOPIES	no. of copies of plot	initiates actual plotting se- quence
	ICODE	paper motion code	

For additional information concerning the Varian  
subroutine library, consult Ref. 2.

VPLOT takes each point in the given array, generates a corresponding x value, scales the values to fit on the given pagesize, and connects them, through the use of subroutine VECT. Once all the points have been connected into a continuous graph, titles and axes are generated, and the entire graph is plotted. Once the plot has been generated, module VPLOT returns control to the PLOTn routine which invoked it. The PLOTn routine, in turn, passes control to NAV320, the driver program.

### 2.3 Optional debug capabilities

At various stages in the processing stream, the ability to actually look at the data is desirable. The user can activate any of the optional debug output statements found in all of the component modules of the NAVELEX surface shipping program. The DEC PDP 11/70 compiler makes it possible to exercise some or all of the debug capabilities built into this program. To activate the debug option in selected modules, the user need only compile the specific module(s) with a /DE option. To exercise all debug options, a single /DE, placed at the end of the list of modules to be compiled, will suffice. Needless to say, indiscriminant use of the debug option can result in an almost unlimited amount of printed output. User discretion is therefore strongly advocated.

Module NAV320, the program driver, has debug statements which issue periodic calls to the system supplied time function. Times are printed at the termination of execution of each module. In this way, it is

possible to monitor the length of time required for processing by each of the component modules.

#### 2.4 Operating Requirements

As was discussed in the introduction to this volume, the NAVELEX Surface Shipping program has been designed for execution on a DEC PDP 11/70 computer. A Floating Point array transform processor and a Varian Statos line printer/plotter were used as peripheral devices. As a result of the limited core size, it was necessary to utilize a number of logical units for temporary storage of intermediate results. The following is a list of the units used and their allocation:

<u>Unit number</u>	<u>Use</u>
1	used by Varian plotter
2	intermediate storage
3	intermediate storage
4	input channel #1
5	card reader
6	line printer
7	input channel #2
8	Floating Point array processor
9	Floating Point array processor
10	Floating Point array processor
11	input unit for filter function
12	intermediate storage
13	intermediate storage
14	output unit for composite covariance functions

The following is a breakdown of the size of each module:

<u>Module name</u>	<u>Size</u>	<u>Size (common blocks excluded)</u>
NAV320	10276	10276
NAV2P	10493	2285
HANG	8414	202
FIN	284	284
NAV34P	8402	197
FILTR	8759	551
PWHITE	903	899
DB	8359	154
SHIFT	8588	383
NAV78P	8632	423
TRANS	8386	181
PLOT1	8440	168
PLOT4	8438	166
PLOT5	9135	863
PLOT9	8503	230
VPLOT	10793	520
		<u>17782</u>

Additional space requirements of the Dataplot III library, the Floating Point library, and usual operating overhead resulted in the task-built unit having size 32K decimal words.

	NAV320	NAV2P	HANG	FIN	NAV34P	FILTR	PWHITE	DB	SHIFT	NAV78P	TRANS	PLOT 1	PLOT 4	PLOT 5	PLOT 9	VPLOT
AA	X	X	X		X	X		X	X	X	X	X	X	X	X	X
NORMAL	X	X	X							X						
BB	X		X		X	X	X	X	X	X	X					
NUMB	X															X
START	X	X	X			X										
LABS												X	X	X	X	X
SF												X	X	X	X	X
IWORK																X

Common Block Allocation

Table 2.1

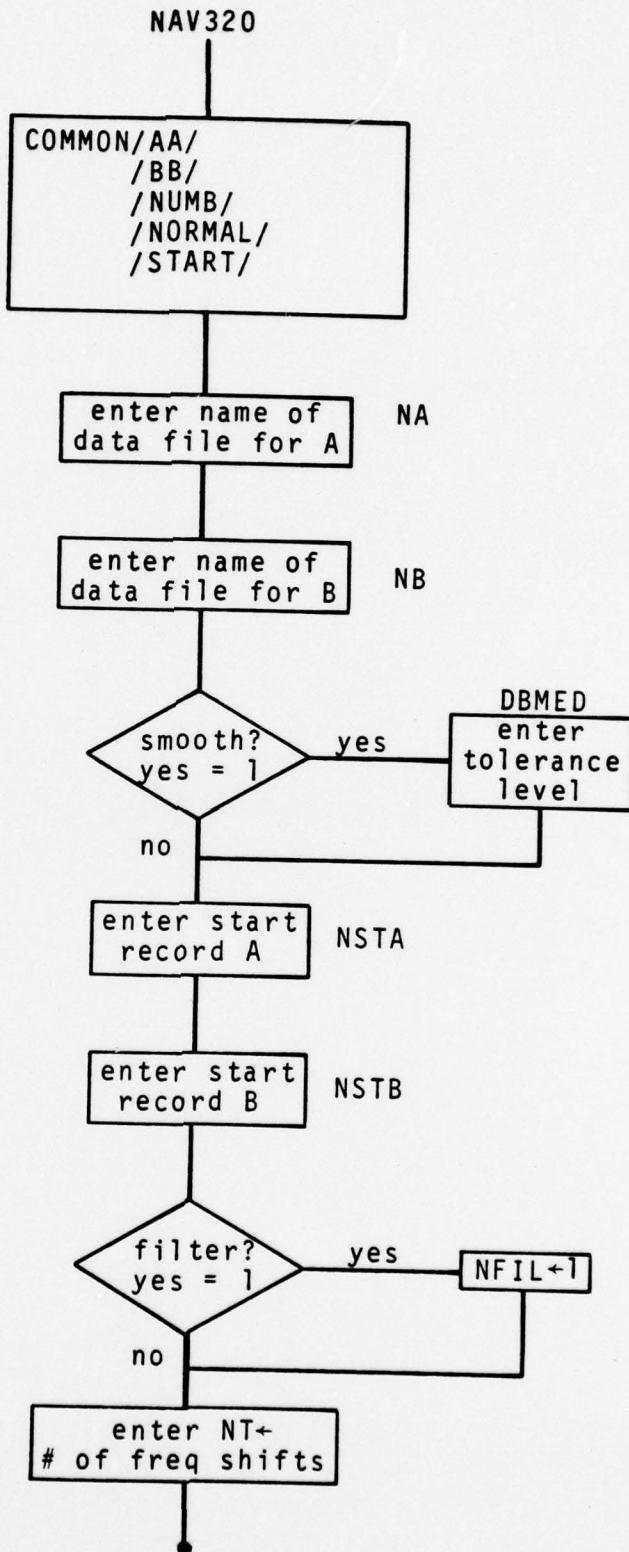


Figure 2.1 Major Flow Pattern

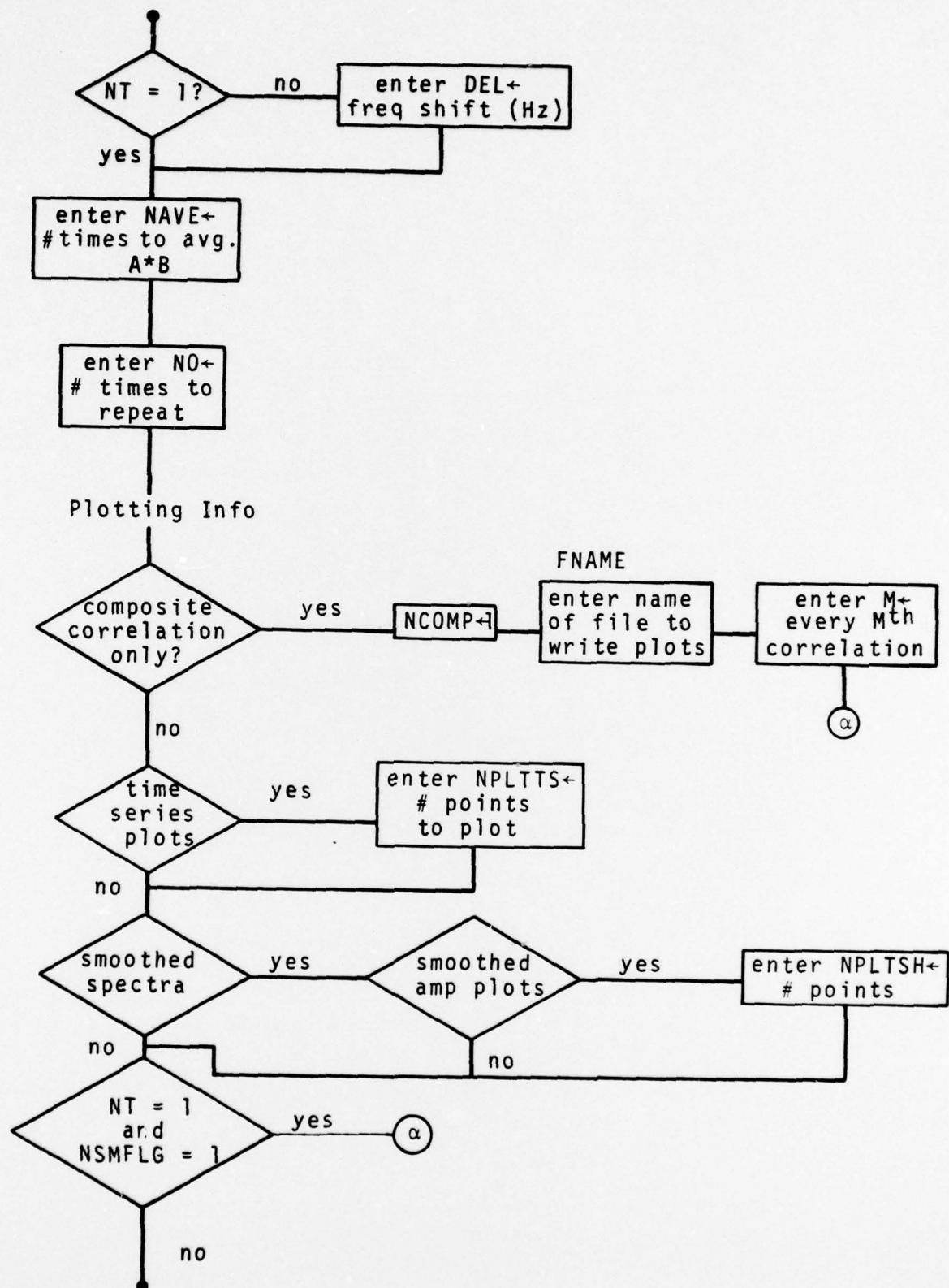


Figure 2.1 Major Flow Pattern (Cont'd)

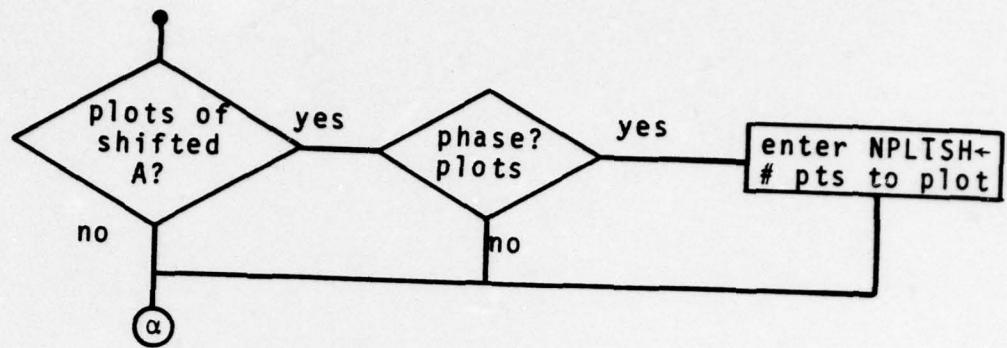


Figure 2.1 Major Flow Pattern (Cont'd)

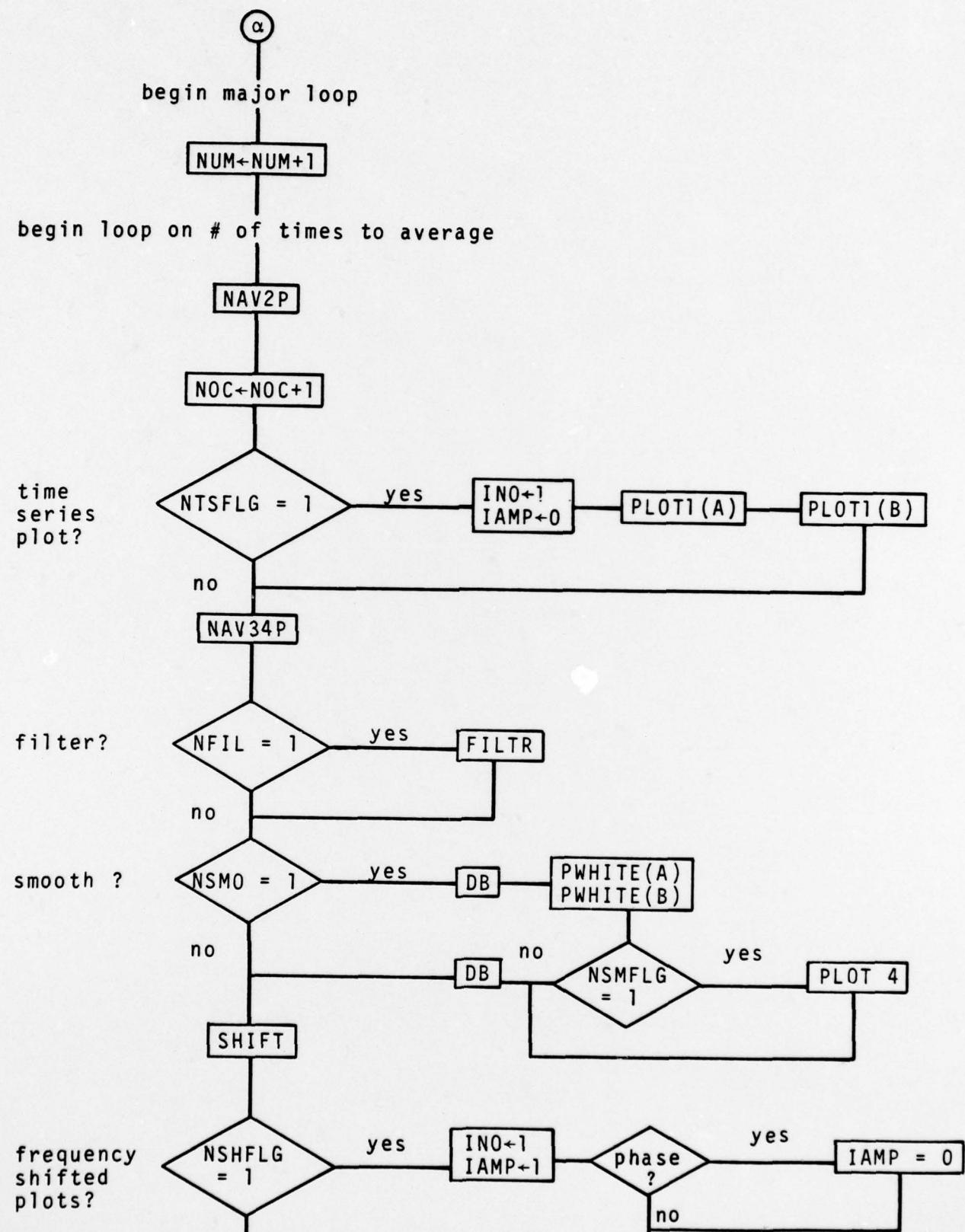


Figure 2.1 Major Flow Pattern (Cont'd)

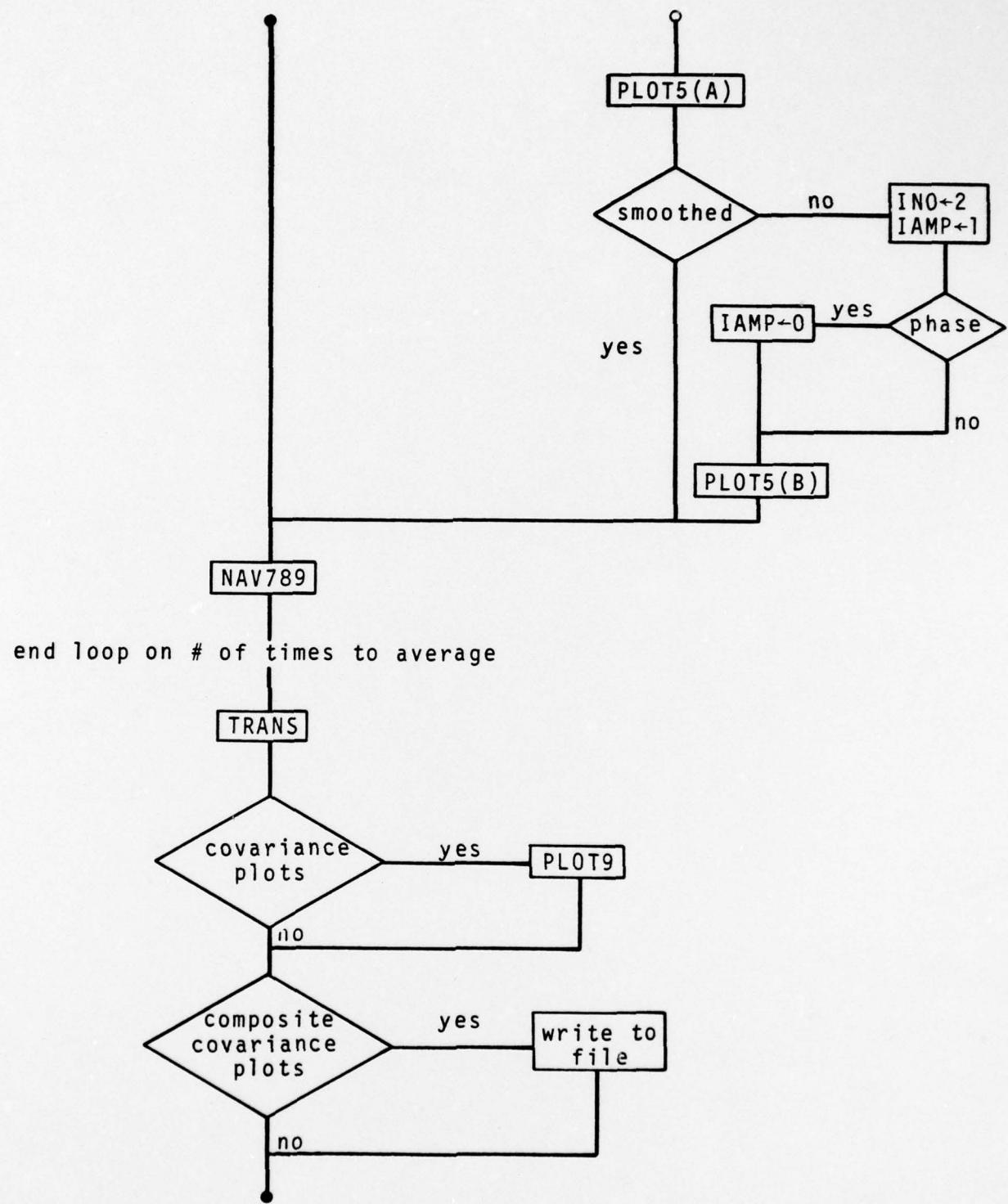


Figure 2.1 Major Flow Pattern (Cont'd)

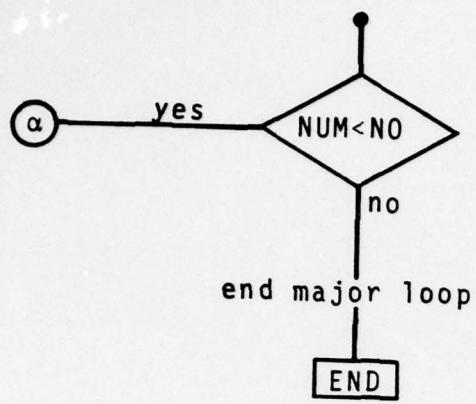


Figure 2.1 Major Flow Pattern (Cont'd)

<u>Common</u>	<u>Variable name</u>	<u>Variable desc</u>
AA	Y(2050,2)	arrays A and B each with 2048 pts
	NOC	case #
LABS	AB1(10)	contains plot title
	AB2(10)	x-axis label
	AB3(10)	y-axis label
	LLAB1	# characters in AB1
	LLAB2	# characters in AB2
	LLAB3	# characters in AB3
	J	frequency shift index
SF	SD	factor for generating x-coordinates
	YSF	calculated y-axis scale factor
	YDISP	calculated y-axis displace- ment factor
SF	NPC	n of PLOTn which issues call to VPLOT; directs certain operations de- pending on value of n
	NUM	frequency shift index
unlabeled	IWORK(2000)	work area for Varian plot- ter; contains machine de- pendent variables

Table 2.2 Common Block Contents  
Plotting Information

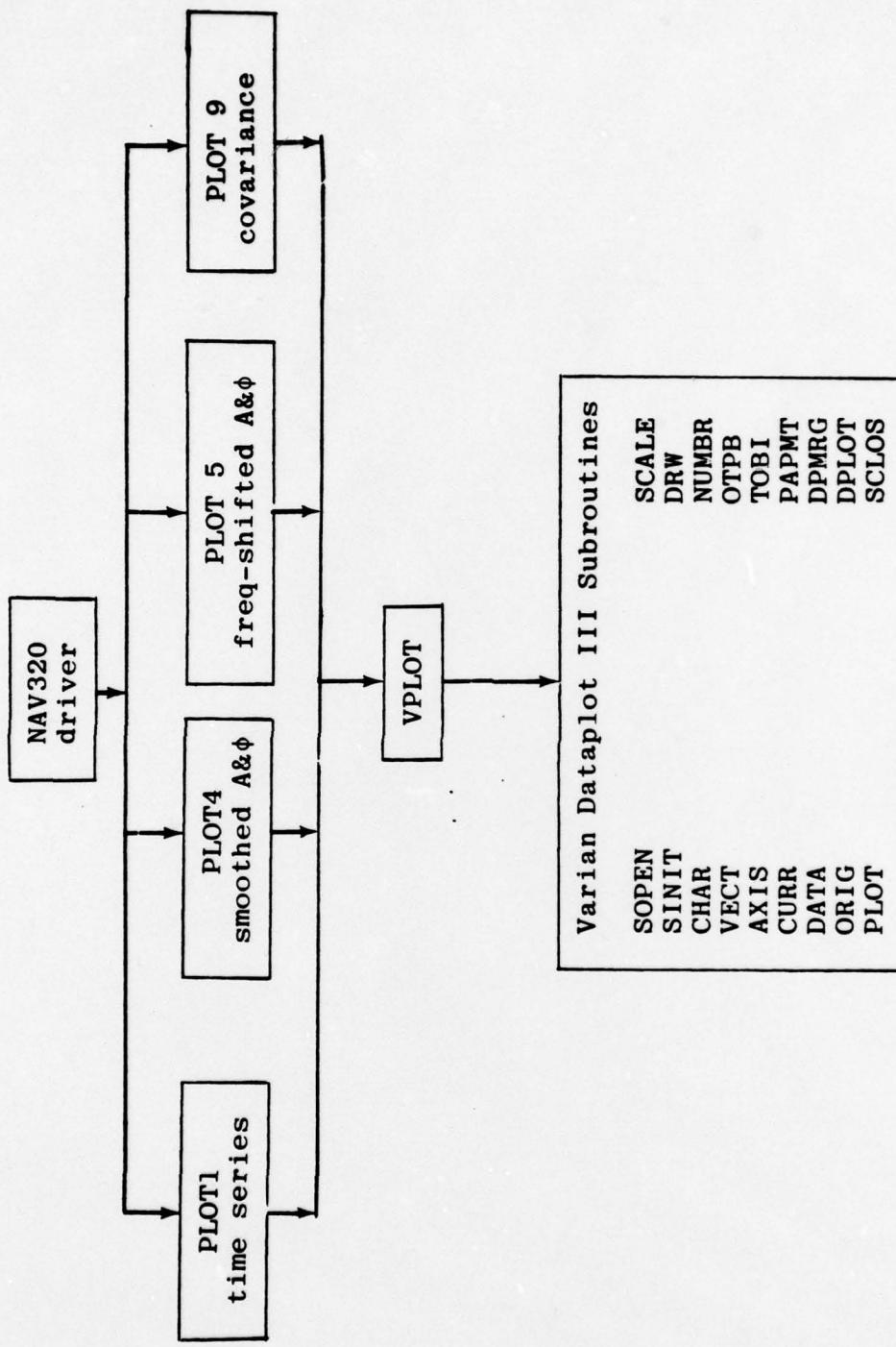


Figure 2.2 Processing flow for plotting routines

#### REFERENCES

1. J. S. Hanna, "Travel-Time Differences from a Broad-band Source for Pairs of Sensors," SAI Tech. Rpt. No. SAI-78-708-WA, 1977.
2. "User's Guide for Dataplot III," Varian Graphics, Pub. No. 03-996284B, 1976.

\* I am indebted to J. S. Hanna, P. V. Rost, R. G. Stieglitz, C. W. Spofford and P. Vodola for their technical assistance.

**Distribution List**

**ARPA Research Center**  
**Unit 1**  
**Moffett Field, CA 94035**  
**ATTN: Dr. Theo Kooij**

**Naval Electronic Systems Command**  
**Department of the Navy**  
**Washington, D.C. 20360**  
**ATTN: ELEX 03, Mr. J. Cybulski**

**Naval Ocean Systems Center**  
**Department of the Navy**  
**San Diego, CA 92152**  
**ATTN: Dr. E. Tunstall**  
**Dr. H. Schenck**

**Defense Documentation Center**  
**Building 5, Cameron Station**  
**Alexandria, VA 22314** (2 cys.)